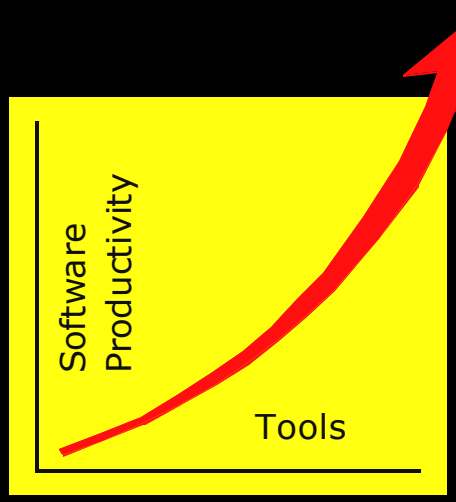
A New Generation of Systematic Programming Tools



James Larus
Software Productivity Tools (SPT)
Programmer Productivity Research Center (PPRC)
Microsoft Research
December 2001

Starting Observation

- No one knows how to write high-quality software in a cost-effective, scalable manner
  - High quality $\Rightarrow$ rich functionality and few errors
  - Cost effective $\Rightarrow$ time and money
  - Scalable $\Rightarrow$ large code base and development team

Lots of Incremental Solutions

- e.g. better practice, technique, language, ...
- Short term answers
  - Current practice dysfunctional in many ways
  - Behavior is entrenched, so small changes easier
- Doomed to fail
  - Messianic religious beliefs
  - Awful academic research

Key Problem

- People are fallible
- Software isn't resilient
- Combination is awful
- One of the most important lessons, perhaps, is the fact that SOFTWARE IS HARD….The creation of good software demands a significantly higher standard of accuracy than those other things do, and it requires a longer attention span than other intellectual tasks. – *Donald Knuth*

So, Let’s Have a Revolution

- Anything is possible
- Alas, no new ideas

SPT

- Exploit (real) revolution
  - Processors: ?40%/yr
  - Disk: ?60%/yr
- Systematic support for developers & testers
  - Capable machines & precise program analysis
- Automatically detect errors and inconsistencies
  - Computers track details
- Improve human-human interaction
- Partial solution
  - Need improvements in design, testing & management